

# Grouped Spatial-Temporal Aggregation for Efficient Action Recognition

Chenxu Luo                      Alan Yuille

Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218, USA

{chenxuluo, ayuille1}@jhu.edu

## Abstract

*Temporal reasoning is an important aspect of video analysis. 3D CNN shows good performance by exploring spatial-temporal features jointly in an unconstrained way, but it also increases the computational cost a lot. Previous works try to reduce the complexity by decoupling the spatial and temporal filters. In this paper, we propose a novel decomposition method that decomposes the feature channels into spatial and temporal groups in parallel. This decomposition can make two groups focus on static and dynamic cues separately. We call this grouped spatial-temporal aggregation (GST). This decomposition is more parameter-efficient and enables us to quantitatively analyze the contributions of spatial and temporal features in different layers. We verify our model on several action recognition tasks that require temporal reasoning and show its effectiveness.*

## 1. Introduction

With the success of convolutional neural networks in image classification [20, 10], action recognition has also shifted from traditional hand-crafted features (e.g. IDT [27]) to deep learning based methods. With the introduction of large scale datasets [19, 3, 8] and more powerful models [3, 29], deep network based methods have become standard for video classification tasks.

Temporal reasoning plays an important role in video analysis. However, common video datasets used for action recognition, such as UCF101 [21] and Kinetics [3], do not require much temporal reasoning. Most of the classes in the datasets can be recognized based only on static scenes or objects [13]. Furthermore, some works even show that shuffling the temporal ordering, the accuracy remains almost the same [33]. This suggests that models trained on those datasets may not necessarily exploit temporal cues.

Recently, several datasets [8, 4, 13] have been released which require temporal modeling. For example, Figure 1 shows two examples from the Something-Something dataset [8]. Seeing only a single frame is not sufficient to determine the class. The two examples are similar at the

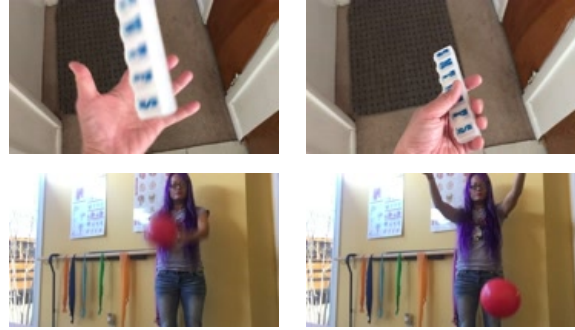


Figure 1. Examples from the something-something dataset [8]. The groundtruth for the two videos are “throwing something in the air and catching it” and “throwing something in the air and letting it fall”. This requires temporal information to correctly differentiate these two classes.

beginning (the first column) but have different results at the end (see the second column). These datasets emphasize on the temporal aspects in action recognition.

However, this does not mean that the static information in each frame is not helpful. Appearance encodes rich cues for temporal reasoning. For example, in Figure 1, we can narrow down the possible interpretations by only seeing a single frame. And we can infer the action from the sparsely sampled frames by observing the state changes.

Existing spatial-temporal networks, such as C3D [25] and I3D [3], learn spatial and temporal features jointly in an unconstrained way. Although they can achieve good performance, they also introduce a large number of parameters that results in computational burdens. Some works [23, 17, 26, 33] try to reduce the cost by decomposing a 3D convolutional kernel into spatial and temporal part separately. However, it remains unsure how spatial and temporal information is utilized in a network.

In this paper, we propose to decompose along the channel dimension instead and show that it is more parameter-efficient than previous methods. Our method is inspired by the widely used group convolutions. The intuition here is that some channels may be more related to spatial features and some channels focus more on motion features,

by analogy to the different functions of neurons (*e.g.* Parvo and Magno cells) in the retina. In previous methods, the spatial and temporal features are entangled together across channels. And directly applying the same operator on all channels may not be optimal and efficient. So we propose to decompose the feature maps into a spatial group and a channel group and apply different operations respectively. Based on this, we design a two-path module in each residual block. Different from previous works where the groups are symmetric, we use one path to model spatial information and the other path to explore temporal information. After that, the spatial-temporal features are concatenated. We call this Grouped Spatial-Temporal aggregation (GST). Unlike the cascaded decomposition used in the P3D-like networks [17], our method implements it in a parallel way, which can exploit features in a more efficient way. This spatial-temporal decomposition not only reduces the parameters but also facilitates the network to learn different aspects (*i.e.* static and dynamic information) and temporal multi-scale features separately in a single layer.

Unlike previous works that model spatial-temporal information in an unconstrained way, our decomposition allows us to analyze how networks exploit spatial and temporal features in different layers. Interestingly, we find that low level features focus more on static cues while high level features focus more on dynamic cues when the networks are trained on temporal modeling tasks. The networks can automatically learn a soft selection without any further constraints.

The proposed module can be easily inserted into any common 2D networks such as ResNet [10]. We conduct experiments on several datasets that require temporal information. Our model can outperform existing methods with less computational cost.

To summarize, our contributions include (a) We propose a novel decomposition method for 3D convolutional kernels that explicitly model spatial and temporal information separately and efficiently; (b) We quantitatively analyze the contribution of spatial and temporal features in different layers; (c) We achieve the state-of-the-art results on several datasets that require temporal modeling with much less computational costs.

## 2. Related Works

**Datasets for Action Recognition** The prevalent datasets such as UCF101 [21] or Kinetics [3] have strong static bias and focus less on temporal orders [13, 33, 35]. Li *et al.* [13] quantitative evaluate the bias towards static representations, such as scenes and objects. Such biases distract researcher from exploring better temporal model. It remains unsure whether the model trained on these datasets actually learn the action itself or simply exploit the bias.

Recently, crowd-acted and fine-grained datasets [8, 19, 4, 7] receive more and more favor and attention. These

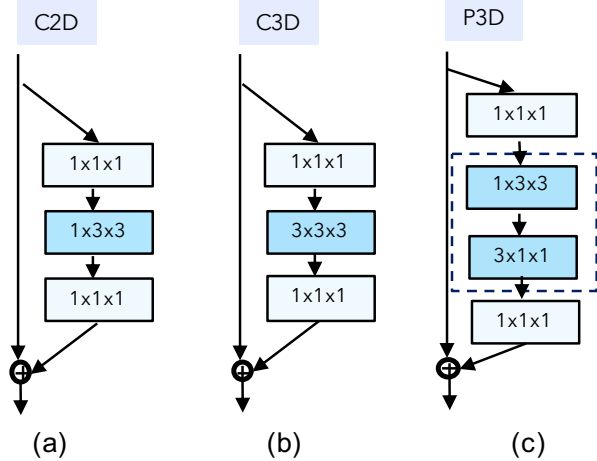


Figure 2. Comparison between three common types of networks. (a) shows a 2D network, TSN [28] and TRN [35] belongs to this category. (b) shows an C3D type network. (c) shows a P3D block (also known as S3D or R(2+1)D), which decouple the spatial and temporal filters.

newly collected datasets pose new challenges for action recognition. Especially fine-grained video datasets such as Something-Something [8, 16], Jester, Diving48 [13] require extensive temporal modeling. For example, the two classes “tearing something into two pieces” and “tearing something just a little bit” in something-something [8] can not be determined without seeing the whole sequence.

**Temporal Modeling** With the success of deep neural networks in visual recognition, a lot of works have been done to extend it for video classification. The early works simply apply 2D convolutions on single frames and fuse them. Karpathy *et al.* [11] propose several fusion strategies for frame aggregation. Later, TSN [28] propose a new sampling strategy and use late fusion strategy to aggregate features of each frame. TRN [35] improves this by introducing multiscale MLP for temporal aggregation. Both of them use late a fusion strategy. Although these 2D networks perform well on datasets like UCF101 [21] or Kinetics [3], they show much less satisfactory results on datasets that require extensive temporal reasoning [8, 13].

In another branch, 3D networks(*e.g.* C3D [25], I3D [3], P3D [17]) recently have gained attention. The first 3D network (*i.e.* C3D [25]) has a huge number of parameters and is hard to train. I3D [3] propose to inflate an ImageNet pre-trained model to 3D by weight copying. Res3D [9] systematically evaluates several common inflated structures. ECO [37] adds a 3D-ResNet after the 2D network for temporal fusion. SlowFast [6] uses two different architectures operating on different temporal frequencies. Our work explores static and motion features in the channel dimension.

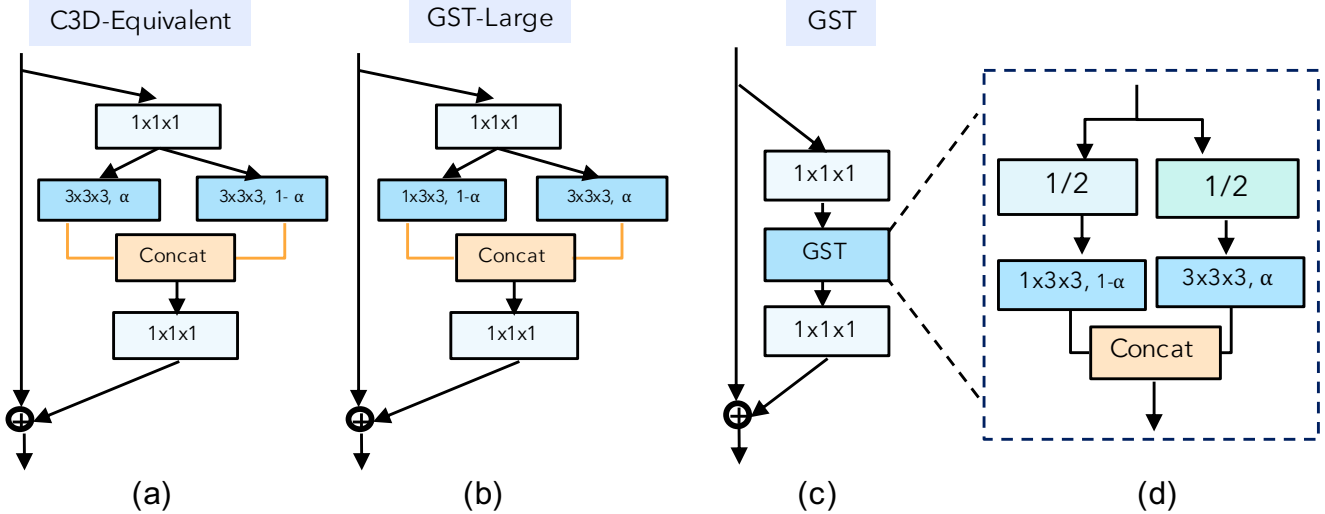


Figure 3. Overview of our proposed method. (a) shows an equivalent network of C3D. (b) shows that replacing one path to spatial only convolutions, denotes as GST-Large. (c) shows our method and (d) illustrates the proposed GST module. In our GST module, the input feature map is divided into two groups; One group for spatial modeling and the other group for temporal modeling. The two paths use the same number of parameters and are concatenated together.

**Optical Flow for Action Recognition** Starting from the seminal work of Two-stream network [20], optical flow has been widely used for motion representation. Most works find their model can perform better when combining optical flow in addition to RGB as input. However, computing optical flow can be time-consuming and is independent of the network. Some works try to jointly optimize optical flow estimation and the classification network [36, 5], or implicitly model optical flow in RGB network [24].

**Efficient Temporal Modeling** Standard 3D networks like C3D [26] contains a huge number of parameters that are difficult to train. Sun *et al.* [23] reduce the parameters by decoupling spatial and temporal kernels. P3D [17] and S3D [32] further explore it with different architectures. R(2+1)D [26] shows that this can achieve better results with the same number of parameters as 3D Convolutions. Figure 2 shows the comparison between these common structures.

TSM [15] replace temporal filters with shift modules. This simple way does not introduce new parameters and can perform surprisingly well on temporal modeling tasks.

### 3. Approach

An overview of our proposed method is shown in Fig.3, the output channels are split into two groups, one for spatial modeling, and the other for spatial-temporal modeling. The spatial part is just the standard 2D convolutions. For the temporal part, 3D convolutions are used. Then the spatial-temporal features are concatenated together. In this way, we

can use even fewer parameters than a standard 2D network counterpart (such as a ResNet-50 [10]) but can significantly boost its ability for temporal modeling. In the following sections, we describe our novel Grouped Spatial-Temporal aggregation (GST) module in detail.

#### 3.1. Decomposing a 3D Convolution Kernel

Consider a 3D convolutional kernel with  $C_i$  input channels and  $C_o$  output channels.  $T, H, W$  are the kernel size along the temporal and spatial dimensions respectively. The kernel is of size  $C_o \times C_i \times T \times H \times W$ , which is  $T$  times larger than its 2D counterpart. Given that modern CNNs such as ResNet [10] usually have a large number of channels, this significantly increases the cost.

There are a lot of works that seek to reduce the parameters by factorizing the convolutional kernels. One widely used way is to decouple the spatial and temporal part [23, 17, 26, 33]. The underlying assumption here is that the spatial and temporal kernels are orthogonal to each other. Mathematically, we can write this decomposition as

$$w = w_t \times w_s \quad (1)$$

where  $w_s \in \mathbb{R}^{C_o \times C_i \times 1 \times H \times W}$  and  $w_t \in \mathbb{R}^{C_o \times C_i \times T \times 1 \times 1}$  are the spatial and temporal kernels respectively. R(2+1)D [26] shows that this decomposition can achieve better performance under the same number of parameters used as 3D convolutions.

### 3.2. Grouped Spatial-temporal Decomposition

Group convolution has been widely used in image recognition, for example, ResNext [32], ShuffleNet [34], to name a few. However, in video tasks, it has been less explored. Most of the existing works simply replace the original convolutions with group convolutions, such as Res3D [9].

However, as shown in the experiments, directly applying group convolutions in a trivial way, which results in symmetric groups, cannot bring improvements. So we propose to decompose the large 3D convolutional filter along the channel dimension in an asymmetric way.

Since both appearance and motion are useful for action recognition, some feature channels may focus more on static appearance while other channels may focus more on dynamic motion features. So modeling them separately is effective and efficient. Based on this assumption, we propose to let the two groups of features model spatial and temporal information separately.

Figure 3 (a) shows an equivalent network architecture to C3D (Figure 2(b)), where the output channels are split into two groups and then concatenated. In our GST design, we apply spatial-only convolutions (*i.e.* 2D convs) to the first group of features and spatial-temporal convolutions (*i.e.* 3D convs) to the other group. We denote this as GST-Large as shown in Figure 3(b). To further reduce the number of parameters, we decompose the input channels into two groups, spatial and temporal, and apply 2D and 3D convolutions respectively. This can encourage the channels in each group to concentrate on static semantic features and dynamic motion features separately and thus easier for training. Static and dynamic features can be thus combined in a natural way. Formally, our decomposition module GST can be written as

$$w_{GST} = w_{gs} \oplus w_{gt} \quad (2)$$

where  $w_{gs} \in \mathbb{R}^{C_{o_s} \times C_{i_s} \times 1 \times H \times W}$  is used for the spatial path and  $w_{gt} \in \mathbb{R}^{C_{o_t} \times C_{i_t} \times T \times H \times W}$  is used for the temporal path. Here,  $o_s$  and  $i_s$  are the number of output and input channels for spatial path, and  $o_t$  and  $i_t$  for the temporal path in the same way. Our method enables multi-scale temporal modeling in a single layer. In the experiments, we show that this spatial-temporal decomposition enjoys better parameter utilization and can effectively reduce the number of parameters while leading to better performance.

### 3.3. Computational Costs for the Spatial and Temporal Path

To control the complexity of the GST, we introduce two parameters to specify the complexity of spatial and temporal branches. We use  $\alpha$  to specify the proportion of temporal output channels and  $\beta$  to specify the number of input channels for spatial and temporal features.

For output channels, we have  $C_{o_t} = \alpha C_o$  number of

channels for the temporal path, and the rest for the spatial path, so the total number of parameters of the spatial and temporal path are:  $(1 - \alpha)HWC_{i_s}C_o$  and  $\alpha THWC_{i_t}C_o$  respectively. Larger values of  $\alpha$  results in more channels for temporal modeling and thus higher computation cost. While smaller  $\alpha$  means lower capacity for temporal path and thus lower complexity. As pointed out in SlowFast [6], lower channel capacity means weaker ability to represent spatial semantics. We carry out experiments with  $\alpha = 1/2, 1/4, 1/8$  respectively. Empirically, we find that fewer temporal channels are beneficial for reducing the computation cost without hurting the performance. In section 4.6, we quantitatively analyze how spatial and temporal channels are utilized in each block.

For input channels, if we set  $\beta = 1$ , then  $C_{i_s} = C_{i_t} = C_i$  and both spatial and temporal path take as input the whole feature maps. We denote this model as GST-Large (Figure 3 (b)). Compared with the model in Figure 3 (a), which is equivalent to 3D convolutions, we replace one path for spatial modeling. This allows multi-scale temporal modeling in a single layer. In the experiments, we show that this not only reduces the parameters, but also improves the performance.

For more efficient architectures, we set  $\beta = 1/2$ , so  $C_{i_s} = C_{i_t} = C_i/2$ . The models are shown in Figure 3 (c) and (d), where the input channels are split evenly into two groups and one group is used for spatial modeling and the other group is used for temporal modeling. With the commonly used kernel size  $H = W = T = 3$ , our GST models have roughly the same or even less number of parameters than a 2D network with properly designed spatial-temporal channel decomposition. However, our model contains sufficient temporal interactions and thus has higher temporal modeling ability than merely using a 2D network.

To summarize, we list the number of parameters for different architectures in Table 1.

Model	# params
C2D	$H \cdot W \cdot C_i \cdot C_o$
C3D	$T \cdot H \cdot W \cdot C_i \cdot C_o$
P3D	$(H \cdot W + T) \cdot C_i \cdot C_o$
C3D(groups=g)	$T \cdot H \cdot W/g \cdot C_i \cdot C_o$
GST-Large	$(1 - \alpha + \alpha T)HWC_iC_o$
GST	$(1 - \alpha + \alpha T)HWC_iC_o/2$

Table 1. Comparison of the number of parameters for each spatial-temporal block.

### 3.4. Network Architecture

The proposed GST module is flexible and can be easily plugged into most of the current networks. More specifically, we replace each of the  $3 \times 3$  convolutional layer with our GST module while keeping other layers unchanged. The final prediction is a simple average pooling of each

frame. We show that this can already achieve good results since the spatial-temporal features are frequently aggregated in each intermediate block. This is contrary to the late fusion method TRN [35], which needs a complex fusion module that operates on the high-level features.

## 4. Experiments

### 4.1. Datasets

We evaluate our method on five video datasets that require temporal modeling.

**Something-Something** Something v1 [8] and v2 [16] are two large scale video datasets for action recognition. There are totally about 110k(v1) and 220k(v2) videos for 174 fine-grained classes with diverse objects, backgrounds, and viewpoints. The fine-grained level classes need extensive temporal reasoning to differentiate them as shown in the example in Fig 1. We mainly conduct experiments and justify each component on these two datasets.

**Diving48** Diving48 [13] is a newly released dataset with more than 18K video clips for 48 diving classes. This requires more focus on pose and motion dynamics. In fact, this dataset aims to minimize the bias towards static frames and facilitate the study of dynamics in action recognition. We report the accuracy on the official train/val split.

**Egocentric Video Datasets** We also evaluate our model on two egocentric video tasks to show that our proposed model is generic on a variety of tasks. We use two recently collected egocentric dataset, Epic Kitchen [4] and EGTEA Gaze+ [14]. For Epic Kitchen, we report the verb classification results using the same split as [1]. EGTEA Gaze++ is a recently collected dataset with approximately 10K samples of 106 activity classes. We use the first split as [14], which contains 8299 training and 2022 testing instances.

### 4.2. Implementation Detail

We implement our model in Pytorch. We adopt ResNet-50 [10] pretrained on Imagenet [18] as the backbone. The parameters of temporal paths are randomly initialized.

For the temporal dimension, we use the sparse sampling method described in TSN [28]. And for spatial dimension, the short side of the input frames are resized to 256 and then cropped to  $224 \times 224$ . We do random cropping and flipping as data augmentation during training time.

We train the network with a batch-size of 24 on 2 GPUs and optimize using SGD with an initial learning rate of 0.01 for about 40 epochs and decay it by a factor of 10 every 10 epochs. The total training epochs are about 60. The dropout ratio is set to be 0.3 as in [30].

During the inference time, we sample the middle frame in each segment and do center crop for each frame. We report the results of **single crop** unless specified.

### 4.3. Results on Something-Something Datasets

We first evaluate each component of our model on both something-something v1 and v2 datasets.

**Ablation Study** We conduct several ablation studies on the Something-Something V1 and V2 validation sets [8]. For all the models, we sample 8 frames using the same sampling method as TSN [28] and use ResNet-50 [10] as backbone network. Results are shown in Table 2.

We compare our model with three baselines, ResNet50 based C3D, C3D with group convolutions and P3D. For C3D and P3D, we use the architecture depicted in Fig. 2 (b) and (c) respectively, and for C3D with groups of 2, we set each  $3 \times 3 \times 3$  convolution to be a group convolution with group size of 2. We also compare networks with different spatial and temporal channel ratios ( $\alpha = 1/2, 1/4, 1/8$  described in Sec.3.3).

Method	#params	v1		v2	
		top1	top5	top1	top5
C3D $_{3 \times 3 \times 3}$	42.5M	46.2	75.6	60.9	87.0
C3D groups=2	29.6M	45.1	74.0	59.9	86.5
P3D	29.4M	45.7	75.0	59.8	85.8
GST-Large(1/4)	29.6M	<b>47.7</b>	<b>76.4</b>	<b>62.0</b>	<b>87.5</b>
C2D	23.9M	20.4	48.1	30.5	61.2
GST ( $\alpha=1/2$ )	23.9M	46.7	<b>76.2</b>	61.4	<b>87.3</b>
GST ( $\alpha=1/4$ )	<u>21.0M</u>	<b>47.0</b>	76.1	<b>61.6</b>	87.2
GST ( $\alpha=1/8$ )	<b>19.7M</b>	46.7	75.7	60.7	86.6

Table 2. Ablation Study on Something v1 and v2 validation set. For all the models, we use a ResNet-50 based backbone and sample 8 frames for each video clip.

First, for our GST-Large model, we set  $\alpha = 1/4$ . This results in a similar number of parameters as P3D or naive 3D group convolutions with a group size of 2. However, our model outperforms other methods on both datasets. Even compared with the larger C3D model, it still performs much better. This shows that our parallel decomposition can better utilize the parameters than the cascaded way like P3D. Also, compared with the original 3D convolutions, GST-Large uses only partial channels for temporal modeling and thus reduces the computational costs significantly. However, our model generalizes better than C3D by decomposing the channel space into spatial and temporal separately.

Second, for more efficient models, our proposed GST uses a similar amount of parameters as a 2D ResNet-50, but performs much better than 2D models. This shows that our model allocates the parameter space more efficiently. Compared with 3D group convolutions, we show that replacing one of the groups with spatial-only convolutions is beneficial. Even compared with the C3D networks, our model combining spatial and temporal cues still performs better on both v1 and v2 dataset. This shows that a 3D network

Model	Backbone	#Frame	GFLOPs	Top1	Top5
TRN-2stream [35]	BN-Inception	8	-	42.0	-
ECO [37]	BNInception+ 3D ResNet-18	8	32	39.6	-
		16	64	41.4	-
MFNet-C50 [12]	ResNet50	10	-	40.3	70.9
MFNet-C101 [12]	ResNet101	10	-	43.9	73.1
NL I3D [29]	3D ResNet-50	32×2 clips	168×2	44.4	76.0
NL I3D+GCN [30]	3D ResNet-50	32×2 clips	-	46.1	76.8
TSM [15]	ResNet-50	8	33	43.4	73.2
TSM [15]	ResNet-50	16	65	44.8	74.5
S3D [33]	BN-Inception	64	66.38	47.3	78.1
S3D-G [33]	BN-Inception	64	71.38	48.2	<b>78.7</b>
GST (ours)	ResNet-50	8	<b>29.5</b>	47.0	76.1
GST (ours)	ResNet-50	8×2 clips	29.5×2	47.6	76.6
GST (ours)	ResNet-50	16	59	<b>48.6</b>	77.9

Table 3. Comparison with state-of-the-art results on the Something V1 validation set. We mainly consider the methods that only take RGB as input for fair comparison. For each model, we report its top 1 and top 5 accuracy as well as its FLOPs.

Method	Frames	Backbone	Val		Test	
			Top-1	Top-5	Top-1	Top-5
TRN [35]	8	BN-Inception	48.8	77.6	50.9	79.3
TSM [15]	8	Resnet-50	59.1*	85.6*	-	-
TSM [15]	16	Resnet-50	59.4*	86.1*	60.4*	87.3*
GST (ours)	8	Resnet-50	61.6	87.2	60.04*	87.17*
GST (ours)	16	Resnet-50	<b>62.6</b>	<b>87.9</b>	<b>61.18*</b>	<b>87.78*</b>
TRN-2stream [35]	8	BN-Inception	55.5	83.1	56.2	83.2
TSM-2stream [15]	16	Resnet-50	63.5	88.6	64.3	90.1

Table 4. Comparison with state-of-the-art results on the something-something v2 dataset. \* denotes results of 5 crops

contains redundancy and empirically, we find that by separating spatial and temporal channels, the networks are easier to train and generalize better.

We also study the impact of temporal channel capacity. We experiment with different temporal channel ratios ( $\alpha = 1/2, 1/4, 1/8$ ). We find that dropping the ratio of temporal channels does not hurt the performance significantly. This shows that maybe lower channel capacity is needed for temporal modeling. In section 4.6, we examine in detail how the temporal channel capacities affect spatial-temporal modeling.

In later experiments, we set  $\alpha = 1/4$  and  $\beta = 1/2$  as default, for its good trade-off between accuracy and efficiency.

**Comparison with state-of-the-arts** The results on v1 and v2 are shown in Table 3 and Table 4 respectively.

On the v1 dataset, our model sampling only 8 frames can already outperform most current methods. Our method outperforms the late fusion method TRN [35] and ECO [37] because it can better encode the spatial and temporal features. Our model can perform as well as S3D using significantly fewer frames and even outperform complex models like non-local network [29] with graph convolution [30].

Compared with v1, v2 is two times larger with fewer label ambiguities. We test it on both validation and test set. Our model again achieves state-of-the-art results. Especially, our single-stream model outperforms two-stream TRN [35] by 5% absolutely. Even though our model takes only RGB as input, our 16-frame model provides competitive results compared with two-stream networks.

#### 4.4. Results on Diving48 Dataset

We test our model on Diving48 [13] dataset. This dataset requires modeling the subtle body motions in order to classify correctly, while the background and object cues seem almost useless. We sample 16 frames from each video clip.

In Table 5, we present quantitative results on this dataset. Compared with previous works, our method outperforms all other counterparts, like R(2+1)D network, by a large margin. Especially, by only employing a lightweight backbone, ResNet-18, our model can already outperform the previous state-of-the-art. This shows that our model can efficiently capture important temporal cues. We believe leveraging pose estimation can benefit recognizing diving actions, but this is beyond the scope of this paper. Despite this, our generic model can already outperform current methods.

Method	Pre-training	Accuracy
C3D(64 frames)(from [14])	-	27.6
R(2+1)D(from [2])	Kinetics	28.9
R(2+1)D+DIMOFS [2]	Kinetics + PoseTrack	31.4
C3D-ResNet18(our impl.)	ImageNet	33.0
P3D-ResNet18(our impl.)	ImageNet	30.8
GST-ResNet18(ours)	ImageNet	<b>34.2</b>
C3D-ResNet50(our impl.)	ImageNet	34.5
P3D-ResNet50(our impl.)	ImageNet	32.4
GST-ResNet50(ours)	ImageNet	<b>38.8</b>

Table 5. Results on the Diving48 Dataset [14]

#### 4.5. Results on Ego-motion Action Recognition

To show that our proposed model is generic for various action recognition tasks, we also test it on two recently released ego-motion video datasets, *i.e.* Epic-Kitchen [4] and EGTEA Gaze++ [14]. Both datasets focus on activities in the kitchen. So there is less bias towards scenes. For the Epic Kitchen Dataset [4], there are a total of 125 verb classes and each verb can be acted on different objects. We report the results on the validation set using the same split as [1]. We only evaluate on the verb class prediction following [1] since the main purpose of this paper is on temporal action recognition instead of objects. For the EGTEA Gaze++ dataset, it contains 106 classes with 19 different verbs. We report the results using the split-1 as in [14].

We use the same setting as the experiments on Something-Something datasets and sample 8 frames for each clip and the results are listed in Table. 6 and 7 respectively.

For the Epic-Kitchen dataset, all models use ResNet-50 as the backbone. Our model again achieves better results.

On EGTEA Gaze++ dataset, we also try a shallow network ResNet-34 as the backbone, for a fair comparison with prior works. Without bells and whistles, our model can even perform better than previous two-stream models with the same backbone architecture. This shows that our proposed module is generic for temporal modeling.

#### 4.6. Analysis of Spatial and Temporal Features

To understand how spatial and temporal information is encoded in each layer, we carefully check the weight of the BN layer after each GST module. The input to the BN layer is a concatenation of spatial and temporal feature maps and the scaling factor of each channel in the BN layer can be used to approximately estimate the importance of that channel. For each bottleneck block, we compute the histograms of the scaling factors of each channel that corresponds to spatial or temporal channel and show them in Figure 4.

The statistics of the scaling factors in the BN layers show that the two groups of channels encode inherently different cues. The network can learn static and dynamic features separately in a single layer and implicitly learn a soft-

Method	[1]	LFB [31]	GST(Ours)
Top1 (Top 5)	40.89 (-)	54.4 (81.8)	<b>56.50 (82.72)</b>

Table 6. Results on validation set of Epic-Kitchen verb classification tasks using the same split as in [1]

Method	Video Acc
[14](I3D-2stream)	53.3
[22](R34-2stream)	62.2
P3D-R34(our impl.)	58.1
GST-R34(ours)	62.2
P3D-R50(our impl.)	61.1
GST-R50(ours)	<b>64.4</b>

Table 7. Results on EGTEA Gaze++ using split 1

weighted dynamic channel selection in each block.

First, in the left column, we show models with different temporal channel ratios  $\alpha$  trained on Something-Something. For  $\alpha = 1/2$ , in block 3, the spatial and temporal weights are less distinguishable, showing that too many temporal channels may encode extra static information. This somehow explains why reducing the number of temporal channels can improve accuracy. While for  $\alpha = 1/8$ , it may not have enough capacity for temporal modeling.

We also visualize the statistic of models trained on Epic-Kitchen, Diving48 and Kinetics. For datasets that require temporal information, we can see that in low-level features, spatial information is more important and in high-level features, temporal information outweighs spatial information. This may due to that object cues in a single frame are often not enough for determining the action. And the temporal channels thus encode abstract motion features other than static features that help recognize actions. However, for Kinetics, spatial and temporal features are less distinguishable. This suggests that the learned temporal features may contain some static features.

Thus, by decoupling spatial and temporal feature channels, we can quantitatively evaluate the contribution of each part. This gives insight into how spatial and temporal cues are encoded from low-level to high-level features, which may benefit future network designs.

We illustrate some examples from Something v2 val set in Figure 5. In each example, we show the network prediction in each intermediate time stamp. Specifically, the final prediction is an average of each frame’s prediction. We examine the output in each intermediate frame. Interestingly, the state transitions can be learned given only video-level labels. In the first example, the prediction goes from “tearing something just a little bit” to “tearing something into two pieces”, which corresponds to the state changes of the whole action. Similarly, the network can change to “pretending to something behind something” after seeing the bottle is moved back. This suggests the state changes of static frames may be crucial to recognize the full action.



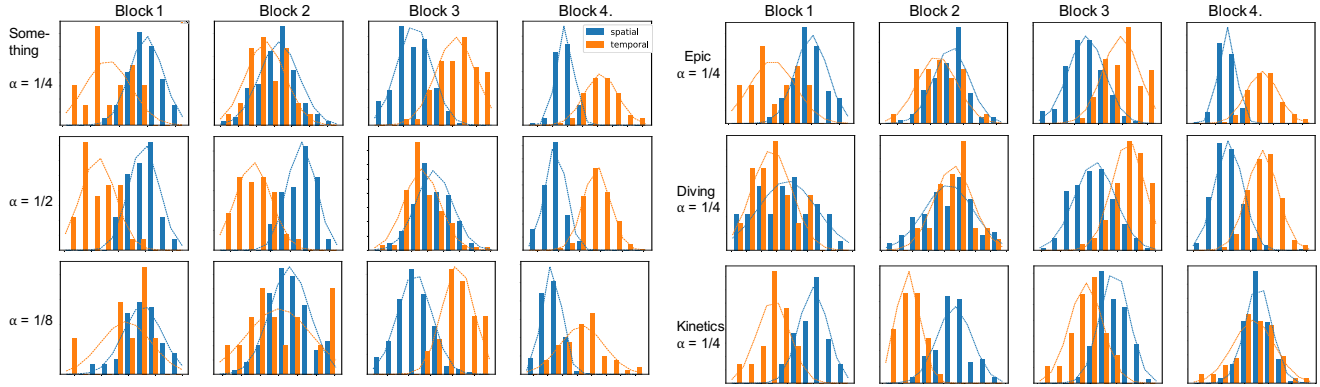


Figure 4. Contributions of spatial and temporal information. We plot the histograms of weights in each BN layer that corresponds to spatial and temporal group respectively after each GST module. Higher weight means the information in that channel is more important.

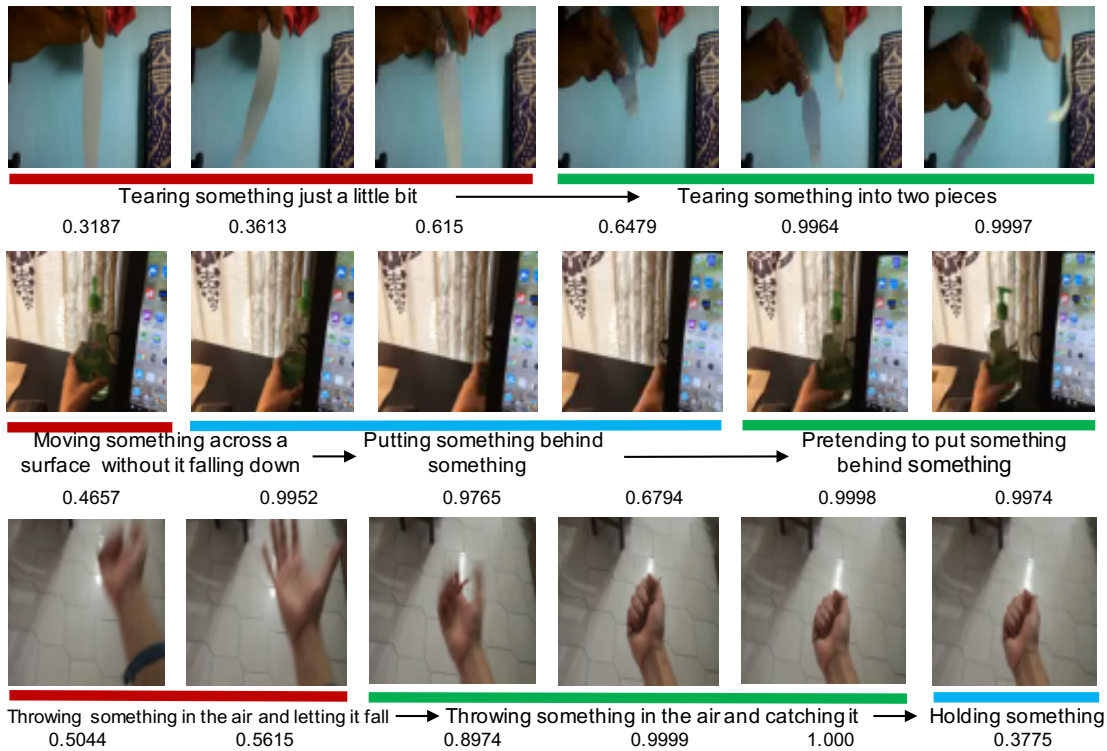


Figure 5. Examples show how the predictions evolve temporally. We use the 16-frame model trained on Something v2 dataset. We only show six typical frames in each video clip. We compute the prediction of each frame before the average pooling and show the predicted label and confidence score for each frame. Green bars show the correct prediction for the whole video clip. Interestingly, state changes can be discovered without strong supervision.

## 5. Conclusions

In this paper, we propose a simple yet efficient network for temporal modeling. The proposed GST module decomposes the feature channels into static and dynamic part, and apply spatial and temporal convolutions separately. This decomposition can effectively decrease the computation cost

and facilitate the network to explore spatial and temporal features in parallel. Further diagnoses give insight into how the two components contribute to the whole network.

**Acknowledgement** This work is partially supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00345.



## References

- [1] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [2] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning discriminative motion features through detection. *arXiv preprint arXiv:1812.04172*, 2018.
- [3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.
- [5] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.
- [7] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4991–5000, 2018.
- [8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The” something something” video database for learning and evaluating visual common sense. In *ICCV*, volume 2, page 8, 2017.
- [9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [12] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018.
- [13] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018.
- [14] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [15] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *arXiv preprint arXiv:1811.08383*, 2018.
- [16] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic. On the effectiveness of task granularity for transfer learning. *arXiv preprint arXiv:1804.09235*, 2018.
- [17] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.
- [18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [19] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016.
- [20] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
- [21] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [22] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *BMVC*, 2018.
- [23] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.
- [24] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [25] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [26] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

- [27] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.
- [28] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.
- [29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.
- [30] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018.
- [31] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. *arXiv preprint arXiv:1812.05038*, 2018.
- [32] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.
- [33] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.
- [34] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.
- [35] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.
- [36] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.
- [37] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 695–712, 2018.